



Hochschule für Technik
und Wirtschaft Berlin

University of Applied Sciences

Kreditkarten-Ausfallprognose durch Anwendung der Ensemble-Methoden Bagging und Boosting

Seminararbeit

MSc FAR WiSe 2023/24

Autor

Sakir Kepezkaya

Datum

Berlin, 20. Februar 2024

Dozentin

Dr. Alla Petukhina

Inhaltsverzeichnis

1	Einleitung	1
2	Methoden	2
2.1	CART	2
2.1.1	Klassifikationsbaum	4
2.2	Ensemble Methoden	6
2.2.1	Bagging	6
2.2.2	Boosting	8
2.3	SMOTE	10
3	Datensatz	12
4	Ergebnisse	14
4.1	Anwendung in Python	14
4.2	Prognosen	16
5	Fazit und Ausblick	22
	Quellen	23
A	Literatur	23
B	Python Pakete	25
	Anhang	26

Zusammenfassung

Diese Arbeit beschäftigt sich mit der Anwendung der Ensemble-Methoden Bagging und Boosting. Dabei wird im Detail untersucht, ob Ensemble-Methoden bessere Ergebnisse für die Kreditkarten-Ausfallprognose liefern können als traditionelle Methoden, wie die logistische Regression, die im Bankenwesen häufig genutzt wird. Zudem werden die Ergebnisse der Ensemble-Modelle miteinander verglichen. Bevor es zur Anwendung kommt, werden beide Modelle aus mathematischer Sicht erörtert. Die Ergebnisse der Vorhersagen konnten schlussendlich als überzeugend eingestuft werden.

Abkürzungsverzeichnis

BaFin Bundesanstalt für Finanzdienstleistungsaufsicht

CART Classification and Regression Tree

KI Künstliche Intelligenz

LIME Local Interpretable Model-Agnostic Explanations

LR Logistische Regression

m Monat

SHAP SHapley Additive exPlanations

SMOTE Synthetic Minority Over-sampling Technique

1 Einleitung

Das Risikomanagement ist nicht nur eine regulatorische Notwendigkeit, das durch die Aufsichtsbehörden vorgesehen wird, sondern ein wichtiges Instrument, um die finanzielle Gesundheit von Unternehmen zu gewährleisten. Im Bankenwesen gibt es verschiedene Risikotypen. Eines dieser Risikotypen ist das Kreditrisiko. Banken gewähren Kredite an Unternehmen sowie Privatpersonen. Dabei besteht das Risiko, dass Kredite nicht zurückgezahlt werden können und Verluste für die Banken entstehen. Aus diesem Grund ist die Beurteilung dieses Risiko notwendig. Eine Möglichkeit zur Beurteilung des Ausfallrisikos ist die Erstellung einer Ausfallprognose.

Die BaFin gibt für diese Art von Modellierungen an, dass die logistische Regression seit Jahrzehnten eine beliebte Methode ist [Kelp and Schneider, 2023]. Jedoch können KI-Methoden genauere Prognosen als traditionelle Methoden liefern und spielen deshalb im Risikomanagement der Banken eine immer größere Rolle [intel, 2024]. Diese Arbeit beschäftigt sich daher mit der Kreditkarten-Ausfallprognose durch die Anwendung der Ensemble-Methoden Bagging und Boosting in Kombination mit einem Entscheidungsbaum. Ferner werden die Ergebnisse mit den einer herkömmlichen Methode, und zwar der logistischen Regression, verglichen.

In der folgenden Abbildung sind die Ergebnisse der Vorhersagen aus dem Artikel [Islam et al., 2018] zu erkennen, welcher denselben Datensatz verwendet hat.





	Extremely Randomized Trees	k-Nearest Neighbor	Gradient Boosting	Naïve Bayes	Random Forest
 F-score	0.9014	0.4862	0.4837	0.3626	0.8637
 Recall	0.8585	0.3687	0.3716	0.4213	0.7932
 Precision	0.9487	0.7134	0.6927	0.3182	0.9478
 Accuracy	0.9584	0.8276	0.8245	0.6723	0.9446

Abbildung 1: Vorhersage-Ergebnisse mit selbem Datensatz aus Artikel [Islam et al., 2018]

So ist zu erkennen, dass die Modelle eine hohe Prognosegenauigkeit haben, aber der Ausmaß je nach Modell variiert. Das beste Ergebnis konnten die „Extremly Randomized Trees“ mit einer Genauigkeit von 0.9584 und einem F-Maß in Höhe von 0.9014 liefern. Außerdem sollte das Boosting-Modell, was auch in dieser Arbeit verwendet wird, hervorgehoben werden. Das Boosting-Modell weist eine Genauigkeit von 0.8245 sowie ein F-Maß von 0.4837 auf und ist somit signifikant schlechter

als die Modelle „Extremly Randomized Trees“ und „Random Forest“.

Jedoch besteht bei der Anwendung von KI-Methoden das Problem der sogenannten „Blackbox“. Durch die komplexe Funktionsweise dieser Modelle können die Modellergebnisse nicht nachvollzogen werden. Weswegen die Anwendbarkeit in der Praxis sich als schwierig gestaltet, da Banken in der Lage sein müssen, die Entscheidungen gegenüber beispielsweise Aufsichtsbehörden rechtfertigen zu können [Bundesbank, 2021, S.8]. Folglich thematisiert diese Arbeit auch die Erklärbarkeit von Modellentscheidungen mithilfe einer Untersuchung der Merkmals-Wichtigkeit.

Zunächst wird im zweiten Abschnitt die angewendete Methodik erläutert. Um die Unausgewogenheit im Datensatz zu adressieren, kommt die SMOTE-Methode zum Einsatz, die speziell für den Ausgleich dieser Datensätze entwickelt wurde. Im dritten Kapitel wird der verwendete Datensatz analysiert und aufbereitet. Die Umsetzung der Modelle ist das Thema des vierten Abschnitts. Anschließend werden die Ergebnisse der Vorhersagen und die Merkmals-Wichtigkeit untersucht. Im letzten Kapitel wird ein Fazit gezogen und ein Ausblick auf mögliche weiterführende Forschungen gegeben.

2 Methoden

2.1 CART

Die Abkürzung CART steht für „Classification and Regression Tree“ und wurde 1984 vom US-amerikanischen Statistiker Leo Breimann eingeführt. Ins deutsche übersetzt heißt es Klassifikations- und Regressionsbaum. CART ist ein Modell, das auf Entscheidungsbäumen basiert. Diese Modelle sind weit verbreitet. Jedoch gibt es weitere baumbasierte Modelle wie z. B. ID3 und C4.5, die aber in dieser Arbeit nicht behandelt werden [Bishop, 2006, S.663].

Es können sowohl Regressions- als auch Klassifikationsprobleme mit dieser Methode gelöst werden. Aus diesem Grund gehört diese Methode zu der Kategorie des Überwachten Lernens.

Beim Maschinellen Lernen existieren verschiedene Ansätze für die Lernalgorithmen. So wird generell zwischen Unüberwachtem und Überwachtem Lernen unterschieden. Der Unterschied besteht darin, dass beim Überwachten Lernen die Daten beschriftet und in Eingabe- sowie Ausgabewerte unterteilt werden. Im Gegensatz dazu besitzt der Datensatz beim Unüberwachten Lernen keine Einteilung und es soll anhand der Zusammenhänge eine Struktur erkannt werden, sodass Cluster

oder Wahrscheinlichkeitsverteilung bestimmt werden können [Goodfellow et al., 2016, S.102 f.].

Das CART-Modell kann durch die sequentielle Entscheidungswahl bei gegebenem Eingabeparameter x definiert werden. Dabei durchlaufen die Entscheidungen einen binären Baum. In der folgenden Abbildung ist ein binärer Baum dargestellt.

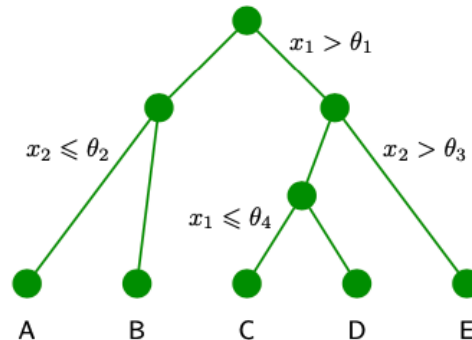


Abbildung 2: Abbildung eines binären Baumes mit zwei Eingabevariablen [Bishop, 2006, S.664]

In einem binären Baum T , wie in Abb. 2, wird zwischen Knoten und Blättern unterschieden. Von einem Knoten gehen genau zwei Kanten aus. Wenn von einem Knoten keine Kanten ausgehen, wird er als Blatt bezeichnet. In der Abbildung wurden die fünf Blätter mit Buchstaben beschriftet. Außerdem wird der erste Knoten auch als Wurzel bezeichnet [Bishop, 2006, S.664 f.].

Basierend auf den Trainingsdaten werden Entscheidungsregeln gesucht und definiert. Anhand dessen werden die Eingabewerte $x = (x_1, x_2, \dots, x_p)$ mit $P = 1, 2, \dots, p$ rekursiv binär aufgeteilt. So entstehen für die Eingabewerte $m = 1, 2, \dots, M$ disjunkte Teilmengen. Diese werden in der Abb. 2 als Blätter dargestellt [Loh, 2011, S.14 f.]. Ein Beispiel für eine Entscheidungsregel ist $x_1 > \theta_1$, wie in der Abb. 2 zu erkennen ist. So wird der Datensatz anhand des Features x_1 mit dem Schwellenwert θ_1 aufgeteilt. Diese Regel wird auch „split rule“ genannt. Dabei ist θ ein Modellparameter und wird auch als „split point“ bezeichnet [Hastie et al., 2009, S.305 ff.]. Die Teilmengen $R = R_1, R_2, \dots, R_M$ stellen die Partition von x dar.

2.1.1 Klassifikationsbaum

Da sich diese Arbeit auf Klassifikationen konzentriert, wird der Algorithmus für einen Klassifikationsbaum betrachtet. Klassifikationen haben kategorielle Daten als Zielvariablen. Im Allgemeinen wird zwischen einer binären Klassifikation und Mehrklassen-Klassifikation unterschieden. Die Klassen werden durch natürliche Zahlen dargestellt, sodass sie Werte $k = 1, 2, \dots, K$ annehmen können. Das Ziel der Klassifikation ist die korrekte Klassenzuordnung der Beobachtung mit einer hohen Wahrscheinlichkeit [Bishop, 2006, S.179 f.].

Die Daten bestehen aus den verschiedenen Eingabeparametern x und den dazugehörigen Ausgabeparameter y für die N Beobachtungen. Dies kann wie folgt dargestellt werden [Hastie et al., 2009, S.307]:

$$(x_i, y_i) \text{ für } i = 1, 2, \dots, N, \text{ mit } x_i = (x_{i1}, x_{i2}, \dots, x_{ip}).$$

Mithilfe einer Verlustfunktion und den Trainingsdaten kann die Struktur des Entscheidungsbaums ermittelt werden.

Bei einer Klassifizierung ist der Gini-Index ein gängiges Maß zur Beurteilung der Leistung eines Entscheidungsbaumes T . Zunächst ist die relative Häufigkeit bezüglich der Klasse k in der Teilmenge R_m mit N_m Beobachtungen wie folgt zu definieren [Hastie et al., 2009, S.309 f.]:

$$\hat{p}_{mk} = \frac{1}{N_m} \sum_{x_i \in R_m} I(y_i = k). \quad (1)$$

Damit beschreibt der Gini-Index ein Maß der Unreinheit in Bezug auf die Klassenzugehörigkeit des jeweiligen Knotens und wird definiert als:

$$Q_m(T) = \sum_{k=1}^K \hat{p}_{mk}(1 - \hat{p}_{mk}). \quad (2)$$

Des Weiteren wird oft die Entropie als Verlustfunktion gewählt, die wie folgt definiert ist:

$$Q_m(T) = \sum_{k=1}^K \hat{p}_{mk} \ln \hat{p}_{mk}. \quad (3)$$

Sowohl der Gini-Index als auch die Entropie nehmen einen kleinen Wert an, wenn die \hat{p}_{mk} nahe 0 oder 1 ist. Die Verlustfunktionen haben den höchsten Wert für $\hat{p}_{mk} = 0.5$ [Bishop, 2006, S.666].

Eine weitere Güte zur Beurteilung der Unreinheit des Knotens ist der Fehlklassifikationsfehler. Diese ist jedoch unterlegen, da sie nicht differenzierbar und somit für numerische Optimierung, wie dem Gradientenverfahren, ungeeignet ist. Darüber hinaus reagieren die ersten beiden Verlustfunktionen empfindlicher auf Veränderungen in der Knotenwahrscheinlichkeit als der Fehlklassifikationsfehler und werden daher bevorzugt [Hastie et al., 2009, S.309 f.].

Dadurch, dass selbst bei einer gegebenen Anzahl an Knoten im Baum die kombinatorische Anzahl sehr groß und rechnerisch kaum lösbar ist, wird ein sogenannter gieriger Algorithmus für die Bestimmung der Baumstruktur verwendet.

1. Start Baum T mit einem Knoten
2. Wahl Eingabeparameter x_p und split point θ , sodass Aufteilung in $R_1(p, \theta) = \{x|x_p \leq \theta\}$ und $R_2(p, \theta) = \{x|x_p > \theta\}$ die Verlustfunktion minimiert
3. Wiederholung Aufteilung an neuen Knoten mit geringstem Verlust
4. Stopp beim Abbruchkriterium, z. B. Mindestanzahl an Beobachtungen in einem Blatt

Zunächst wird mit einem Knoten gestartet (1) und es werden weitere Knoten nacheinander hinzugefügt (3). Daher muss für jeden Schritt die jeweiligen Teilmengen R_m durch die Wahl der Eingabevariable für die Entscheidungsregel und den split point θ bestimmt werden (2). Dies wird mithilfe der Verlustfunktionen durchgeführt. Folglich wird der Unreinheitsmaß für jede Möglichkeit berechnet. Bei jedem Schritt wird die Eingangsvariable und der Schwellenwert so ausgewählt, dass der Gini-Index oder die Kreuzentropie minimiert wird. Die jeweilige Mehrheitsklasse stellt den besten Schätzer für die Klassenvorhersage einer Region R_m dar.

Der gierige Algorithmus sieht vor, dass die Zugabe weiterer Knoten abgebrochen werden muss. Dabei wird oft die Anzahl an Datenpunkten, die in einer Teilmenge R_m vorhanden sind, als Abbruchkriterium verwendet (4) [Bishop, 2006, S.664 ff.].

Der entstehende Entscheidungsbaum T wird als Nächstes gestutzt, d. h., dass innere Knoten entfernt werden. Dafür wird das „Cost-Complexity-Pruning“ verwendet. Ein Teilbaum wird definiert als $T \subset T_0$, wobei T einen Entscheidungsbaum darstellt, der durch das Stutzen des Baumes T_0 entsteht. Daher ist das Cost-Complexity-Pruning definiert durch [Hastie et al., 2009, S.308]:

$$C_\alpha(T) = \sum_{m=1}^{|T|} N_m Q_m(T) + \alpha |T|. \quad (4)$$

Der Parameter α bestimmt je nach Wahl die Komplexität und die Anpassungsgüte. Dieser Parameter soll so bestimmt werden, dass ein Teilbaum $T_\alpha \subset T_0$ gefunden wird, der $C_\alpha(T)$ minimiert. Hierbei repräsentiert der Betrag von T die Anzahl der Blätter. Bei $\alpha = 0$ entspricht der Baum dem vollen Baum T_0 , sodass die Komplexität des Baumes für größere α sinkt. Der Wert dieses Parameters wird durch die Kreuzvalidierung geschätzt und der entstehende Baum wird angegeben als $T_{\hat{\alpha}}$. Diese Methode eignet sich, um die Überanpassung bei Entscheidungsbäumen zu kontrollieren [Hastie et al., 2009, S.308].

2.2 Ensemble Methoden

2.2.1 Bagging

Der in Abschnitt 2.1.1 vorgestellte Entscheidungsbaum hat den Nachteil, dass eine hohe Abhängigkeit zu den Trainingsdaten besteht. Kleine Änderungen können zu unterschiedlichen split points führen und somit in einer zu hohen Varianz in der Entscheidungsregel resultieren. Zudem können diese Bäume schlecht additive Modellstrukturen abbilden. Zur Vermeidung dieser Nachteile können Methoden wie Bagging und Boosting angewendet werden. Diese sind sogenannte „Ensemble Learning“-Techniken. Dies bezeichnet das Kombinieren mehrerer einfacher Modelle, um die Prognosegenauigkeit sowie die Volatilität zu verbessern [Hastie et al., 2009, S.605].

Bagging ist die Abkürzung für „Bootstrap aggregation“ und beschreibt das Erzeugen von Bootstrap-Stichproben, woraufhin dann neue Bäume B erstellt werden, sowie die Kombination der Ergebnisse der neuen Bäume. Die Bootstrap Methode beschreibt das Ziehen mit Zurücklegen aus der Gesamtheit. Das Bagging führt zur Reduktion der Varianz [Hastie et al., 2009, S.282].

Die Aggregation der Ergebnisse erfolgt bei einem Klassifikationsproblem oft über die Mehrheitsentscheidung. So wird für eine Beobachtung x_i die Klasse als Vorhersage gewählt, die am häufigsten für die B Bäume getroffen wurde. Hierfür wird eine Indikatorfunktion

$$\hat{f}^b(x) = \begin{cases} 1, & \hat{y}_b = k, \\ 0, & \hat{y}_b \neq k. \end{cases} \quad (5)$$

verwendet, die den Wert 1 annimmt, wenn die Klasse k für einen Baum vorhergesagt wurde, und ansonsten 0. Darüberhinaus kann der Bagging Schätzer wie folgt definiert werden [Hastie et al., 2009, S.283]:

$$\hat{f}_{bag}(x) = \sum_{b=1}^B \hat{f}^b(x). \quad (6)$$

Schließlich kann die Vorhersage mit der Mehrheitsentscheidung dargestellt werden als:

$$\hat{G}_{bag}(x) = \arg \max_k \hat{f}_{bag}(x). \quad (7)$$

Eine weitere Möglichkeit ist das Bagging über die Wahrscheinlichkeiten, dass eine Klasse k eintritt. So wird die Bagging-Klassenwahrscheinlichkeit über den Durchschnitt der Klassenwahrscheinlichkeiten für die B erzeugten Bäume angegeben. Die Wahrscheinlichkeit des jeweiligen Baumes ist gegeben durch $\hat{f}_k^b(x)$ und der Schätzer für die Bagging-Klassenwahrscheinlichkeit wird definiert als [Hastie et al., 2009, S.283]:

$$\hat{f}_k^{bag}(x) = \frac{1}{B} \sum_{b=1}^B \hat{f}_k^b(x). \quad (8)$$

Außerdem sollte erwähnt werden, dass der Parameter B , die Anzahl der Bäume, bei einer hohen Zahl nicht zur Überanpassung führt. Eine Überanpassung, auch „Overfitting“ genannt, beschreibt, dass das Modell sich zu stark am Trainingsdatensatz orientiert und dabei eine Generalisierung für noch nicht gesehene Datensätze misslingt. Daher sollte diese Variable groß genug gewählt werden, um

eine gute Fehlerminimierung zu erreichen [James et al., 2023, S.344].

2.2.2 Boosting

Die Boosting-Methode hat denselben Zweck wie die Bagging-Methode. Mit ihr soll eine bessere Prognosegenauigkeit mit einer niedrigeren Varianz erzielt werden. Ähnlich wie beim Bagging-Ansatz wird eine Vielzahl an Bäumen erzeugt. Jedoch werden diese Bäume nicht unabhängig voneinander sondern sequentiell generiert [James et al., 2023, S.347 ff.]. In der folgenden Grafik werden die beiden Methoden visualisiert.

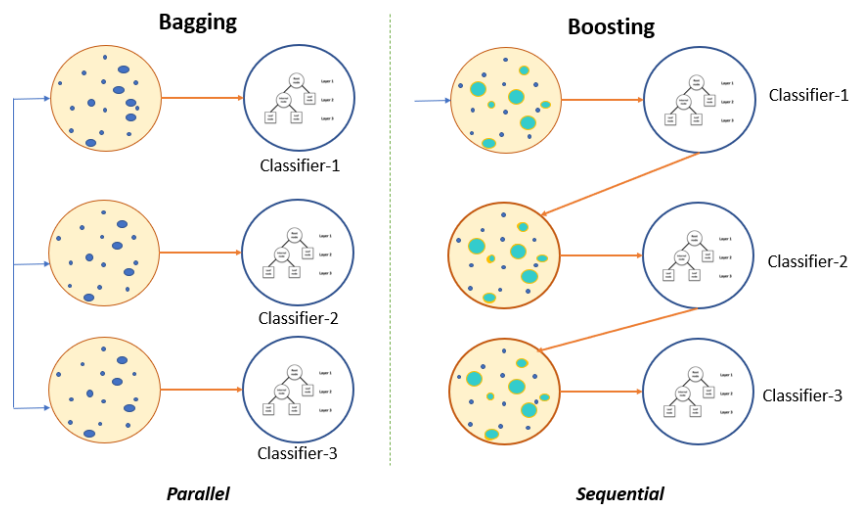


Abbildung 3: Vergleich zwischen den Methoden Bagging und Boosting [Hong, 2021]

Wie in der Abb. 3 zu erkennen ist, wird der Baum auf dem Vorwissen der vorherigen Bäume erzeugt. Die Idee dahinter ist, dass der Baum ausgehend von einem Basisklassifikatoren, der auch „weak classifier“ genannt wird, langsam lernt. In dem Kontext dieser Arbeit beschreibt der „weak classifier“ einen einfachen Entscheidungsbaum mit wenig Knoten [Hastie et al., 2009, S.347].

Im Folgenden wird der Boosting-Algorithmus „AdaBoost“ betrachtet, der 1995 von Yoav Freund and Robert Schapire vorgestellt wurde. Hierfür wird zunächst ein binäres Klassifikationsproblem mit $Y \in \{1, -1\}$ angenommen. Für das Training des Entscheidungsbaumes ist die Berechnung des Fehlers notwendig. Der Fehler des

eines Modells $G(x)$ zur Vorhersage der Klasse y_i wird definiert als:

$$err = \frac{1}{N} \sum_{i=1}^N I(y_i \neq G(x_i)). \quad (9)$$

Folgendermaßen wird der Baum sequentiell für jeden Klassifikator $G(x)$ mit $m = 1, 2, \dots, M$ trainiert und dabei werden die Beobachtungen (x_i, y_i) mit $i = 1, 2, \dots, N$ gewichtet. Die Gewichte w_i werden nach jeder Sequenz m angepasst und das Startgewicht ist definiert als $w_i = \frac{1}{N}$. Der Algorithmus lautet wie folgt.

1. Initialisieren der Gewichte der Beobachtungen $w_i = \frac{1}{N}$ mit $i = 1, 2, \dots, N$
2. Für $m = 1$ bis M
 - (a) Anpassen des Klassifikators $G_m(x)$ an die Trainingsdaten mit Hilfe der Gewichte w_i
 - (b) Berechnung des Fehlers

$$err_m = \frac{\sum_{i=1}^N w_i I(y_i \neq G(x_i))}{\sum_{i=1}^N w_i} \quad (10)$$

- (c) Berechnung des Parameters

$$\alpha_m = \log\left(\frac{1 - err_m}{err_m}\right) \quad (11)$$

- (d) Aktualisieren des Gewichts

$$w_i \leftarrow w_i \cdot \exp[\alpha_m \cdot I(y_i \neq G(x_i))] \quad (12)$$

3. Ausgabe

$$G(x) = \text{sign}\left[\sum_{m=1}^M \alpha_m G_m(x)\right] \quad (13)$$

So wird nach Festlegung der Startgewichte der Klassifikator $G_m(x)$ mit den Gewichten w_i angepasst. Anschließend wird der gewichtete Fehler, wie in Gleichung (10), berechnet.

Ein weiterer wichtiger Parameter in Adaboost ist α_m . Dieser wird für jedes Modell $G_m(x)$ mithilfe des Fehler berechnet, wie in Gleichung (11) zu erkennen ist. Dieser

Parameter entscheidet den Beitrag von jedem Klassifikator. Hiermit werden genauere Klassifikatoren höher gewichtet.

Nach Berechnung des Fehlers und des Parameters α_m eines Klassifikators werden die Gewichte w_i aktualisiert. Das Ziel dieser Anpassung der Gewichte ist, dass der Fokus auf den Beobachtungen liegt, die falsch vorhergesagt wurden. Dadurch werden die Gewichte w_i der falsch vorhergesagten Beobachtungen mit Hilfe der Exponentialfunktion, wie in Gleichung (12), erhöht und für die richtigen Vorhersagen reduziert. Dies hat zur Folge, dass der Einfluss der falsch vorhergesagten Beobachtungen steigt und sich der Klassifikator bei der nächsten Iteration auf diese konzentriert.

Die Anpassung des Modells $G_m(x)$ an die Gewichte w_i , die Berechnung der Fehler und des Parameters a_m des Modells $G_m(x)$ sowie die Aktualisierung der Gewichte w_i der Beobachtungen werden M mal mit $m = 1, 2, \dots, M$ iteriert [Hastie et al., 2009, S.337 ff.]. Dabei ist M ein Parameter des Boosting Modells, der festgelegt werden muss. So werden entsprechend M Klassifikatoren bzw. Entscheidungsbäume erstellt. Jedoch soll hierbei beachtet werden, dass eine zu hohe Anzahl an M , ungleich wie beim Bagging, zur Überanpassung führen kann [James et al., 2023, S.349].

Anschließend wird das Gesamtmodell durch eine gewichtete Summe der einzelnen schwachen Klassifikatoren, wie in Gleichung (13), definiert. Dabei ist $sign()$ die Vorzeichenfunktion. Dies führt dann wiederum zu einer Klassenzuordnung [Hastie et al., 2009, S.337 ff.].

2.3 SMOTE

Um mit unausgewogenen Datensätzen umgehen zu können, gibt es verschiedene Methoden. Zwei häufig verwendete Ansätze sind die „Oversampling“- und „Undersampling“-Techniken. Beim Undersampling wird die Anzahl der Mehrheitsklasse reduziert, um die Klassen auszugleichen. Daraus resultiert, dass der Datensatz insgesamt schrumpft und Informationen verloren gehen. Wiederum wird beim Oversampling die Minderheitsklasse durch das Hinzufügen neuer Beobachtungen erhöht, um die Klassen auszugleichen. Dies hat zur Folge, dass der Datensatz größer wird. Jedoch steigt dadurch auch der Rechenaufwand [Mayuri S. Shelke1, 2017].

In dieser Arbeit wird die Oversampling-Technik „Synthetic Minority Over-sampling Technique“ (SMOTE) verwendet. SMOTE erhöht die Minderheitsklasse durch die Erzeugung synthetischer Beobachtungen. In der folgenden Grafik wird die Funktionsweise visualisiert.

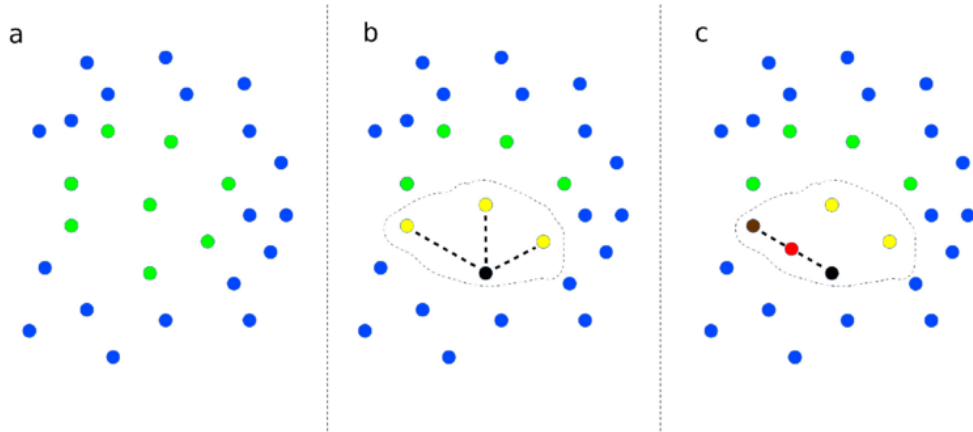


Abbildung 4: Synthetic Minority Over-sampling Technique [Schubach et al., 2017]

In Abb. 4 wird in (a) die Minderheitsklasse mit grünen Punkten repräsentiert. In (b) wird dann ein zufälliger Punkt aus der Minderheitsklasse ausgewählt und eine bestimmte Anzahl an Punkten in der Nähe mithilfe des Algorithmus „k-nearest neighbors“ (k-NN) ausgesucht. In dem Beispiel der Abb. 4 wurden drei sogenannte Nachbarn identifiziert. Anschließend werden zu den Nachbarn Linien gezogen. Im letzten Schritt wird durch Interpolation die synthetische Beobachtung erzeugt. Das bedeutet, der Abstand zwischen dem Nachbarn und der zufälligen Beobachtung mit einem Faktor zwischen 0 und 1 multipliziert wird. So stellt der rote Punkt in (c) eine synthetische Beobachtung dar [Mayuri S. Shelke1, 2017].

3 Datensatz

In dieser Arbeit wurde der Datensatz [Cheng Yeh, 2009] verwendet. Dieser beinhaltet Informationen zu Kreditkartenkunden aus Taiwan. Die Zielvariable ist die Information, ob eine Kreditkarte im nächsten Monat ausfällt. Dadurch ergibt sich ein binäres Klassifikationsproblem. Die Daten wurden in einem Zeitraum von April bis September 2005 erhoben. Insgesamt sind 30.000 Beobachtung mit 23 Merkmalen in dem Datensatz vorhanden. In der folgende Tabelle ist eine Übersicht der Merkmale abgebildet.

Merkmal	Ausprägung
Kreditkartenlimit	10.000 bis 1.000.000
Geschlecht	1 = männlich, 2 = weiblich
Bildung	1 = Hochschule, 2 = Universität, 3 = Oberschule, 4 = andere, 5 = n.a., 6 = n.a.
Familienstand	1 = verheiratet, 2 = ledig, 3 = andere
Alter	21 bis 79 Jahre
Ausgaben1 bis Ausgaben6	-339.603 bis 1.664.089
Einkommen1 bis Einkommen6	0 bis 1.684.259
Rückzahlungsstatus1 bis Rückzahlungsstatus6	-2 = n.a., -1 = fristgerecht, 0 = n.a., 1 = 1 m Verzug, ..., 8 = 8 m Verzug, 9 = ≥ 9 m Verzug

Tabelle 1: Merkmalsübersicht

Die Beträge sind in der taiwanesischen Währung angegeben, dem Neuen Taiwan-Dollar. Die kategorischen Merkmal „Geschlecht“, „Bildung“, „Familienstand“, „Rückzahlungsstatus1“ bis „Rückzahlungsstatus6“ sind bereits ordinal encodiert. In der Tabelle 1 (Abschnitt 3) sind die jeweiligen Werte für die Kategorien zuerkennen. Jedoch fällt auf, dass einige Werte nicht bekannt sind. Dabei sind vor allem die Merkmale „Rückzahlungsstatus1“ bis „Rückzahlungsstatus6“ hervorzuheben. Denn über die Hälfte der Werte werden den unbekannten Kategorien -2 und 0 zugeordnet.

Darüber hinaus sind die letzten drei Merkmale in der Tabelle 1 (Abschnitt 3) jeweils für die sechs Monate des Erhebungszeitraums vorhanden. So repräsentiert z.

B. das Merkmal „Ausgabe1“ die Ausgaben im Monat September 2005, „Ausgabe2“ die Ausgaben im Monat August 2005, ..., „Ausgabe6“ die Ausgaben im Monat April 2005. Das gleiche Schema der Notation gilt auch für die Merkmale „Einkommen1“ bis „Einkommen6“ und „Rückzahlungsstatus1“ bis „Rückzahlungsstatus6“. In der folgenden Visualisierung ist die Verteilung der Zielvariable „Kreditkarten-Ausfall im nächsten Monat“ zu erkennen.

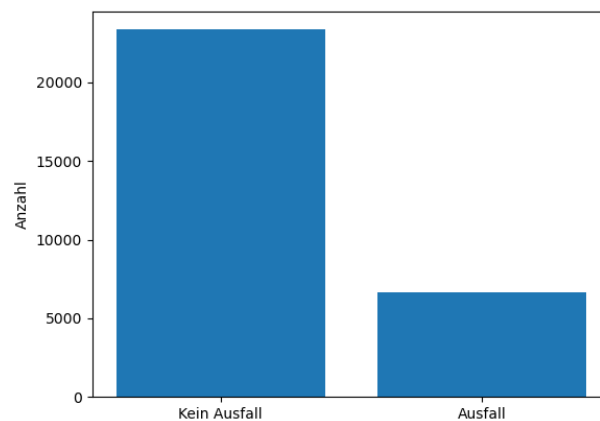


Abbildung 5: Verteilung des Kreditkarten-Ausfalls

So ist zu entnehmen, dass die Zielvariable nicht balanciert ist, d. h., es besteht ein Klassenungleichgewicht. Es gibt deutlich weniger Ausfälle. Insgesamt sind 23.364 Fälle ohne Ausfall und 6.636 Ausfälle vorhanden, was einem Verhältnis von etwa 7:2 entspricht. Somit fallen 22,12 % der Kreditkarten im nächsten Monat aus.

4 Ergebnisse

4.1 Anwendung in Python

Zunächst wurden die Daten mithilfe der SMOTE-Methode (Abschnitt 2.3) ausgeglichen. Dabei war das Ziel, eine verbesserte Prognosegenauigkeit für die Minderheitsklasse zu erreichen, ohne dabei die Gesamtprognosegenauigkeit stark zu beeinträchtigen. So wurde der Datensatz zu einem Verhältnis von etwa 7:2 zu 5:3 balanciert.

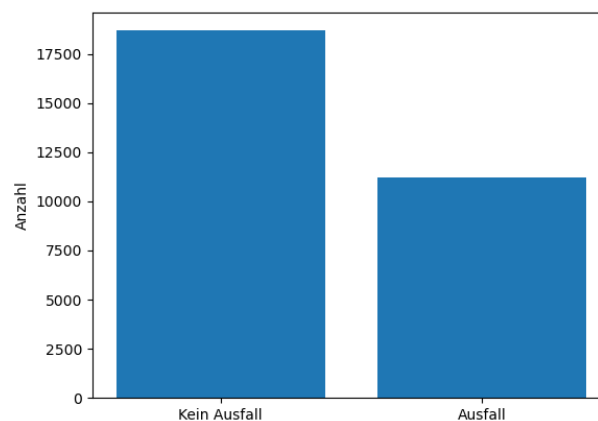


Abbildung 6: Verteilung des Kreditkarten-Ausfalls

Anschließend wurden ein Bagging-Klassifikator und ein AdaBoost-Klassifikator erstellt. Die Modelle wurden mithilfe des Paketes „scikit-learn“ implementiert. Wie bereits erwähnt, werden für diese Art von Modellprognosen in der Finanzwelt oft die logistische Regression verwendet. Aus diesem Grund wurde eine logistische Regression als Benchmark-Modell genutzt. Alle Modelle wurden für den unbalancierten und balancierten Datensatz entwickelt.

Eine geeignete Wahl der Hyperparameter ist notwendig, um eine hohe Prognosegenauigkeit zu erreichen. Daher wurden die Hyperparameter der Modelle mithilfe der Funktion „GridSearchCV“ optimiert. Hierfür wird zunächst ein Satz von möglichen Parametern definiert. Im folgenden Abschnitt sind die Parameterraster der Ensemble-Methoden abgebildet.

Code: Parameterraster

```
# Bagging-Klassifikator
param_grid_bag = {
    'n_estimators': [100, 200],
    'max_samples': [1.0, 0.8],
    'max_features': [1.0, 0.8],
    'base_estimator__max_depth': [1, 5, 10],
    'base_estimator__min_samples_leaf': [1, 5, 10],
    'random_state': [7]
}

# Boosting Klassifikator
param_grid_ada = {
    'n_estimators': [100, 200],
    'base_estimator__max_depth': [1, 5, 10],
    'base_estimator__min_samples_leaf': [1, 5, 10],
    'learning_rate': [0.01, 0.1, 0.5],
    'random_state': [7]
}
```

So durchläuft die Funktion „GridSearchCV“ alle möglichen Parameterkombination und bewertet sie unter Anwendung des Kreuzvalidierungsverfahrens anhand der Genauigkeit. Die Hyperparameter umfassen beispielsweise die Anzahl der Bäume „n_estimators“ oder die Mindestanzahl an Blättern in einem Knoten „min_samples_leaf“. Dabei sollte beachtet werden, dass die Hyperparameter sich je nach Modell unterscheiden können. So hat z. B. der Boosting-Klassifikator zusätzlich den Parameter „learning_rate“, dieser entspricht der Gleichung (11). Das Ergebnis der Hyperparameter-Optimierung lautet wie folgt.

Code: Hyperparameter-Auswahl

```
## Bagging-Klassifikator
# unbalancierte Daten
{100, 1.0, 0.8, 5, 10}
# balancierte Daten
{200, 1.0, 0.8, 10, 1}

## Boosting Klassifikator
# unbalancierte Daten
{200, 5, 10, 0.01}
# balancierte Daten
{200, 10, 1, 0.1}
```

Es ist beispielsweise zu erkennen, dass alle Ensemble-Modelle, abgesehen vom Bagging-Klassifikator mit unbalancierten Daten, 200 Bäume verwenden.

4.2 Prognosen

Im folgenden Abschnitt wird nun analysiert, wie hoch die Prognosegenauigkeit des Kreditkarten-Ausfalls im nächsten Monat ist. Anfangs wird das Ergebnis des Benchmark-Modells für die unbalancierten Daten betrachtet.

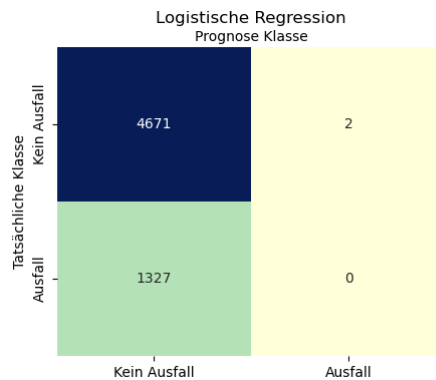


Abbildung 7: Konfusionsmatrix unbalancierten Daten

In der Konfusionsmatrix werden auf der Hauptdiagonalen die korrekt vorhergesagten Klassen angezeigt. Die logistische Regression hat beinahe alle korrekten Fälle der Mehrheitsklasse „Kein Ausfall“ richtig prognostiziert, die sogenannten „True Positiv“-Werte. Jedoch war das LR-Modell nicht in der Lage, Ausfälle zu erkennen. Das LR-Modell hat bis auf zwei Ausnahmen, wo es zusätzlich falsch lag, immer die Mehrheitsklasse prognostiziert. In der folgenden Grafik sind die Prognosen der Ensemble-Methoden mit den unbalancierten Daten abgebildet.

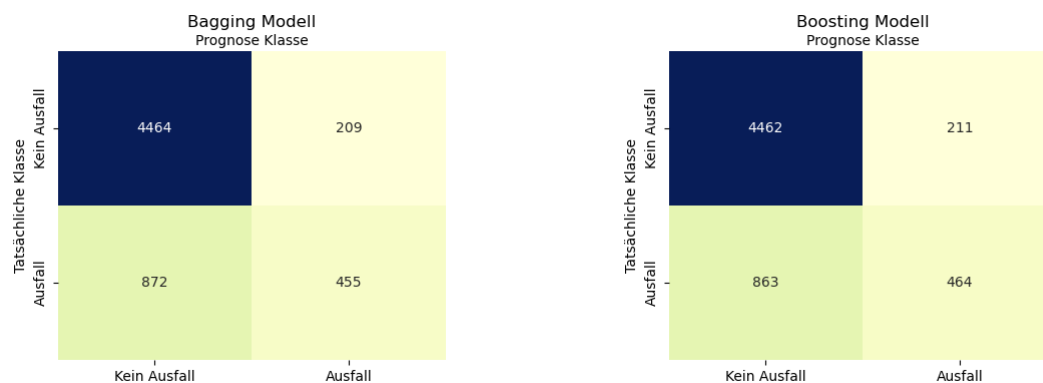


Abbildung 8: Konfusionsmatrix unbalancierten Daten

Es ist ersichtlich, dass beide Modelle sehr ähnliche Ergebnisse erzielten. Im Vergleich zum Benchmark-Modell konnten die Ensemble-Methoden die Mehrheitsklasse geringfügig schlechter vorhersagen. Jedoch waren die Ensemble-Methoden in der Lage, die Minderheitsklasse zu erkennen. Das Bagging-Modell hat 455 von insgesamt 1282 Ausfällen richtig vorhergesagt, während das Boosting-Modell mit 464 eine etwas höhere Anzahl an sogenannten „True Negativ“-Werten erreichte. Im Folgenden werden die Konfusionsmatrizen der Modelle mit balancierten Daten betrachtet.

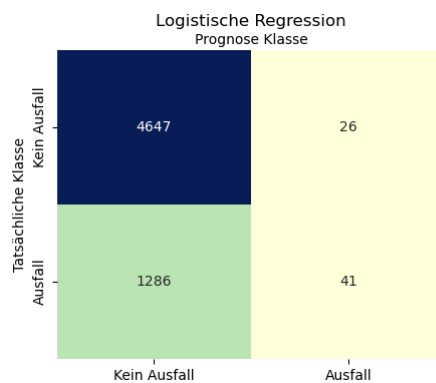


Abbildung 9: Konfusionsmatrix balancierte Daten

Das Ausgleichen der Daten zum Verhältnis von 5:3 hatte für die logistische Regression zur Folge, dass richtige Fälle für die Minderheitsklasse vorhergesagt werden konnten und zwar gibt es 41 „True Negativ“-Werte. Jedoch führte dies auch zu einer geringfügigen Verschlechterung der Prognose für die Mehrheitsklasse.

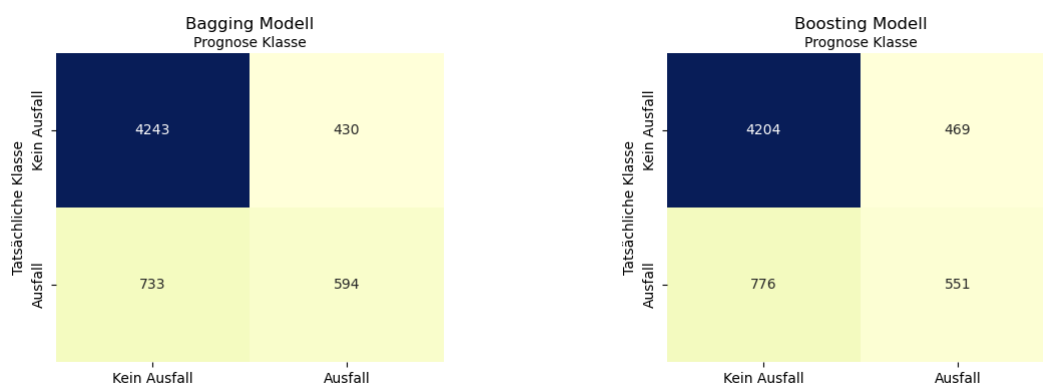


Abbildung 10: Konfusionsmatrix balancierte Daten

Dieselbe Beobachtung gilt auch für die Ensemble-Modelle mit balancierten Daten. Die „True Positiv“-Werte sind für beide Modelle gesunken. Wiederum ist die Anzahl der korrekt vorhergesagten Fälle für die Minderheitsklasse gestiegen. Für das Bagging-Modell konnte ein Anstieg für die „True Negativ“-Werte von 455 auf 594 verzeichnet werden. Im Gegensatz dazu ist beim Boosting-Modell die Anzahl der „True Negativ“-Werte von 464 auf 551 gestiegen. Das bedeutet, dass es für die „True Negativ“-Werte des Bagging-Modell einen höheren Anstieg gab, sodass das Bagging-Modell die meisten richtigen Fälle der Minderheitsklasse korrekt vorhersagt hat.

Im Folgenden werden nun verschiedene Metriken zur Beurteilung der Ergebnisse betrachtet.

Modell	Klasse	Präzision	Sensitivität	F-Maß	Genauigkeit
Logistische Regression	Kein Ausfall	0.7788	0.9996	0.8756	0.7785
	Ausfall	0	0	0	
Bagging	Kein Ausfall	0.8366	0.9553	0.8920	0.8198
	Ausfall	0.6852	0.3429	0.4571	
Boosting	Kein Ausfall	0.8379	0.9549	0.8926	0.8210
	Ausfall	0.6874	0.3497	0.4635	

Tabelle 2: Metrik der Modelle mit unbalancierten Daten

In der Tabelle 2 (Abschnitt 4.2) sind für die drei Modelle mit balancierten Daten die Präzision, die Sensitivität und das F-Maß für beide Klassen abgebildet. Außerdem wird in der letzten Spalte die Genauigkeit für jedes Modell angegeben. Die besten Werte wurden blau markiert in der Tabelle 2 (Abschnitt 4.2). Die Präzision betrachtet innerhalb einer Klasse den Prozentsatz der korrekt positiv klassifizierten Instanzen. Wiederum gibt die Sensitivität für jede Klasse den Prozentsatz der korrekt klassifizierten tatsächlich positiven Instanzen an. Das F-Maß ist das gewichtete Mittel der eben beschriebenen Kennzahlen [sci, 2019]. So ist vor allem das F-Maß der Minderheitsklasse „Ausfall“ und die Genauigkeit von großer Bedeutung. Denn eine Bank strebt bei einer Ausfallprognose vor allem die Identifizierung der korrekten Ausfälle an. Die besten Werte dieser beiden Kennzahlen wurden in roter Schrift hervorgehoben.

In der Tabelle 2 (Abschnitt 4.2) können die Beobachtungen der Konfusionsmatrizen erneut festgestellt werden. Das LR-Modell ist nicht geeignet, da es keine

Instanzen der Minderheitsklasse vorhersagen konnte. Die Ensemble-Methoden haben für die Kennzahlen sehr ähnliche Ergebnisse. Allerdings wies das Boosting-Modell für beide Klassen eine geringfügig höheres F-Maß sowie eine insgesamt höhere Genauigkeit in Höhe von 0.8210 auf. Somit erreichte das Boosting-Modell die besten Ergebnisse unter den Modellen mit unbalancierten Daten. Als Nächstes werden die Metriken der Modelle mit balancierten Daten betrachtet.

Modell	Klasse	Präzision	Sensitivität	F-Maß	Genauigkeit
Logistische Regression	Kein Ausfall	0.7833	0.9944	0.8763	0.7813
	Ausfall	0.6119	0.0309	0.0588	
Bagging	Kein Ausfall	0.8527	0.9080	0.8795	0.8062
	Ausfall	0.5801	0.4476	0.5053	
Boosting	Kein Ausfall	0.8442	0.8996	0.8710	0.7925
	Ausfall	0.5402	0.4152	0.4695	

Tabelle 3: Modelle mit balancierten Daten

Auch für die balancierten Daten kann, wie bereits bei der Konfusionsmatrix zu erkennen war, festgehalten werden, dass die logistische Regression keine zufriedenstellenden Vorhersageergebnisse für die Minderheitsklasse liefert. So konnten mithilfe der balancierten Daten Ausfälle korrekt identifiziert werden, jedoch ist das F-Maß mit einem Wert von 0.0588 nicht signifikant. Das F-Maß der Ensemble-Methoden konnte im Vergleich zu den Modellen mit unbalancierten Daten eine Verbesserung verzeichnen. Den höchsten Anstieg mit etwa 5 % Punkten erzielte das F-Maß des Bagging-Modells. Diese Veränderungen waren aber mit Auswirkungen auf die Genauigkeit der Ensemble-Modelle verbunden, sodass sie um ca. 2 % gesunken ist. Die besten Ergebnisse liefert insgesamt das Bagging-Modell mit einem F-Maß der Minderheitsklasse von 0.5053 und einer Genauigkeit in Höhe von 0.8062.

Als nächstes wird die Merkmals-Wichtigkeit der Modelle untersucht. In den folgenden Abbildungen sind die zehn Merkmale mit der größten Gewichtung der Ensemble-Modellen visualisiert.

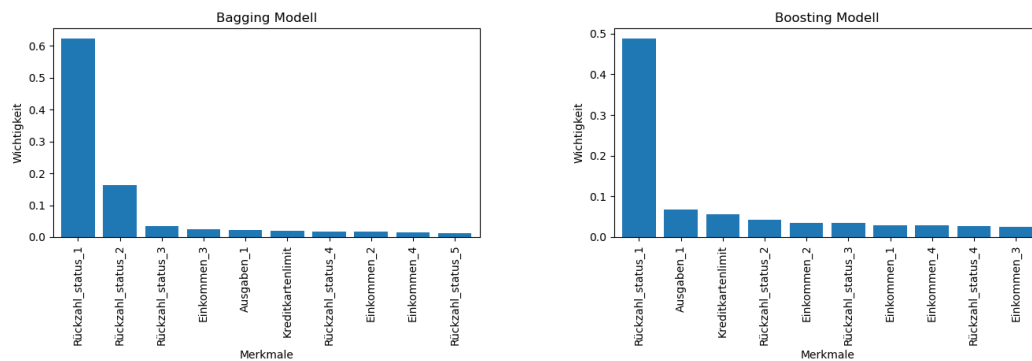


Abbildung 11: Modelle mit unbalancierten Daten

Zunächst werden die Modelle mit unbalancierten Daten analysiert. So ist zu erkennen, dass vor allem ein Merkmal zur Entscheidungsfindung herangezogen wird. Dieses Merkmal ist für beide Ensemble der Rückzahlungsstatus im letzten Monat „Rückzahl_status_1“. Dies konnte auch bei der logistische Regression beobachtet werden (siehe, Abb. 13). Jedoch ist beim LR-Modell insbesondere das Merkmal „Alter“ von Bedeutung.

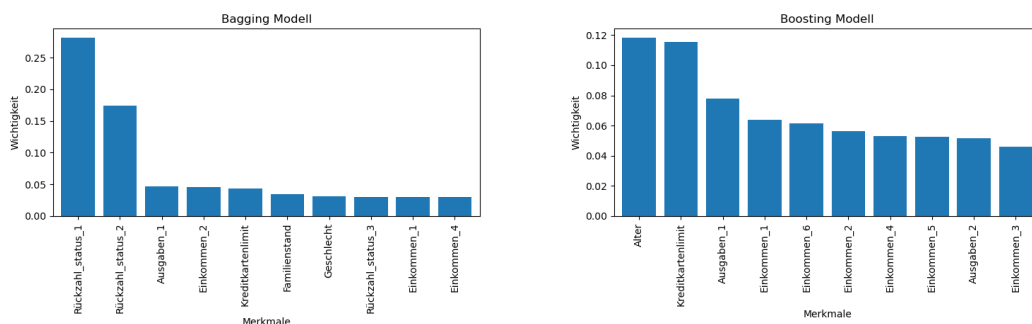


Abbildung 12: Modelle mit balancierten Daten

Das Balancieren der Daten hatte für die Merkmals-Wichtigkeit des Bagging-Modells keine signifikante Auswirkung. Weiterhin ist vor allem das Merkmal „Rückzahl_status_1“ entscheidend. Jedoch ist der Koeffizient dieses Mermales um etwa die Hälfte gesunken. Somit beruht die Entscheidungsfindung nicht so stark auf einem Merkmal wie bei den unbalancierten Daten. Diese Entwicklung konnte für das Boosting-Modell noch stärker beobachtet werden. Außerdem ist auch eine Veränderung der wichtigsten Merkmale zu beobachten. So sind besonders das „Alter“ und das „Kreditkartenlimit“ entscheidend. Auch für

das LR-Modell ist eine Streuung der Entscheidungsfindung auf mehrere Merkmale zu erkennen (siehe, Abb. 14). Zudem ist nicht mehr das Alter das wichtigste Merkmal, wie beim LR-Modell mit unbalancierten Daten, sondern das Einkommen im letzten Monat.

Allerdings muss bei der Analyse der Merkmals-Wichtigkeit beachtet werden, dass, wie bereits in Abschnitt 3 beschrieben, viele Werte des Merkmals „Rückzahl_status_1“ unbekannt sind und dies die Interpretation der Modellentscheidung erschwert.

Abschließend werden die Ergebnisse der Boosting-Modelle dieser Arbeit mit den Ergebnisse des Boosting-Modells aus dem Artikel [Islam et al., 2018] verglichen.

Modell	Klasse	Präzision	Sensitivität	F-Maß	Genauigkeit
Adaboost unbalanciert	Ausfall	0.6874	0.3497	0.4635	0.8210
Adaboost balanciert	Ausfall	0.5402	0.4152	0.4695	0.7925
Gradient Boosting (Artikel)	Ausfall	0.6927	0.3716	0.4837	0.8245

Tabelle 4: Vergleich Kennzahlen der Boosting Modelle mit Ergebnis aus Artikel [Islam et al., 2018]

Im Artikel wurde das „Gradient Boosting“-Verfahren verwendet und in dieser Arbeit der Adaboost-Algorithmus. Außerdem wurden die Daten im Artikel nicht balanciert. In der Tabelle sind die bereits zuvor verwendeten Metriken für die Minderheitsklasse aufgeführt. Daraus geht hervor, dass die Ergebnisse sehr ähnlich sind. Insbesondere das Adaboost-Modell mit unbalancierten Daten hat sehr ähnliche Ergebnisse wie das Boosting-Modell aus dem Artikel. Dennoch konnte das Modell aus dem Artikel im Vergleich zu den in dieser Arbeit verwendeten Modellen eine marginale Verbesserung in Bezug auf das F-Maß und die Genauigkeit erzielen.

5 Fazit und Ausblick

In dieser Arbeit wurden die Ensemble-Methoden Bagging und Boosting verwendet, um eine Kreditkarten-Ausfallprognose durchzuführen. Das Ziel war eine höhere Prognosegenauigkeit als traditionelle Methoden, die seit vielen Jahren eine Anwendung im Bankenwesen haben, zu erzielen. Aus diesem Grund wurden die Ergebnisse der Ensemble-Modelle mit den Ergebnisse einer logistische Regression verglichen. Die Untersuchung beinhaltet somit die Analyse der drei Modelle. Außerdem wurden die Daten ausgeglichen. Dadurch wurden für jedes Modell zwei Datensätze verwendet, sodass insgesamt sechs Modelle miteinander verglichen wurden.

Zusammenfassend lässt sich festhalten, dass die Ensemble-Methoden eine bessere Prognose als die logistische Regression erzielten. Außerdem erreichten die Modelle mit balancierten Daten bessere Prognoseergebnisse für die Minderheitsklasse als die Modelle mit unbalancierten Daten. Jedoch war dies mit einer marginalen Verschlechterung der Genauigkeit verbunden. Zudem sollte erwähnt werden, dass das LR-Modell für die Identifizierung der Ausfälle ungeeignet ist, da es entweder keine Ausfälle oder nur eine sehr geringe Anzahl an Ausfällen erkennen konnte. Die Ensemble-Modelle Bagging und Boosting erwiesen insgesamt ähnlich gute Ergebnisse. Dabei war das Boosting-Modell für die unbalancierten Daten geringfügig besser als das Bagging-Modell hinsichtlich der Metriken F-Maß und Genauigkeit. Im Gegensatz dazu führte die Balancierung der Daten dazu, dass das Bagging-Modell eine etwas bessere Prognose als das Boosting-Modell liefern konnte. In Bezug auf die Merkmals-Wichtigkeit kann zusammengefasst werden, dass zum einen durch die Balancierung der Daten die Entscheidungsfindung nicht hauptsächlich auf einem Merkmal beruhte und zum anderen, dass vor allem die Merkmale Alter und der Rückzahlungsstatus im letzten Monat relevant bei der Modellentscheidung sind. Allerdings sind viele Werte des Rückzahlungsstatus im letzten Monat unbekannt, was die Interpretation der Modellentscheidung beeinträchtigt.

In zukünftigen Forschungen könnte z. B. die Integration eines Random Forest in Betracht gezogen werden. Da, wie im Vergleich zum Artikel [Islam et al., 2018] zu erkennen ist, dieses Modell bessere Prognosen liefert als ein Boosting-Modell, welches ähnliche Ergebnisse wie das Boosting-Modell dieser Arbeit erzielte. Zudem ist das Durchführen einer Merkmalsselektion sinnvoll, um die Komplexität der Modelle und somit Modellinterpretation zu vereinfachen. Zusätzlich ist die Anwendung von „Explainable AI“-Methoden wie SHAP und LIME geeignet, um mehr Aufschlüsse über die Modellentscheidungen zu erhalten [Gramegna and Giudici, 2021].

Quellen

Literatur

- [sci, 2019] (2016 - 2019). Classification Report, letzter Zugriff: 28.12.2023. *scikit-yb.org*. https://www.scikit-yb.org/en/latest/api/classifier/classification_report.html.
- [Bishop, 2006] Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer Science+Business Media, LLC, New York.
- [Bundesbank, 2021] Bundesbank (2021). Maschinelles Lernen in Risikomodellen – Charakteristika und aufsichtliche Schwerpunkte. *bundesbank.de*.
- [Cheng Yeh, 2009] Cheng Yeh (2009). default of credit card clients. *UCI Machine Learning Repository*. <https://archive.ics.uci.edu/dataset/350/default+of+credit+card+clients>.
- [Goodfellow et al., 2016] Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press.
- [Gramegna and Giudici, 2021] Gramegna, A. and Giudici, P. (2021). Shap and lime: An evaluation of discriminative power in credit risk. *Frontiers in Artificial Intelligence*.
- [Hastie et al., 2009] Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The Elements of Statistical Learning*. Springer New York.
- [Hong, 2021] Hong, S. (2021). Ensemble – Bagging and Boosting, letzter Zugriff: 31.12.2023. *seongjuhong.com*. <https://seongjuhong.com/2021-01-17pm-ensemble-bagging-and-boosting/>.
- [intel, 2024] intel (2024). Risikomanagement im Bankwesen, letzter Zugriff: 05.01.2024. *intel.de*. <https://www.intel.de/content/www/de/de/financial-services-it/banking/banking-risk-management.html>.
- [Islam et al., 2018] Islam, S. R., Eberle, W., and Ghafoor, S. K. (2018). Credit Default Mining Using Combined Machine Learning and Heuristic Approach. *CoRR*.
- [James et al., 2023] James, G., Witten, D., Hastie, T., Tibshirani, R., and Taylor, J. (2023). *Introduction*. Springer International Publishing.

- [Kelp and Schneider, 2023] Kelp, T. and Schneider, M. (2023). When an algorithm decides on the loan, letzter Zugriff: 25.12.2023. *www.bafin.de*. https://www.bafin.de/SharedDocs/Veroeffentlichungen/EN/Fachartikel/2023/fa_bj_2305_Algorithmen_Kreditvergabe_en.html.
- [Loh, 2011] Loh, W. (2011). Classification and Regression Trees. *WIREs Data Mining and Knowledge Discovery*.
- [Mayuri S. Shelke1, 2017] Mayuri S. Shelke1, Prashant R. Deshmukh, V. K. S. (2017). A Review on Imbalanced Data Handling Using Undersampling and Oversampling Technique. *International Journal of Recent Trends in Engineering and Research*.
- [Schubach et al., 2017] Schubach, M., Re, M., Robinson, P. N., and Valentini, G. (2017). Imbalance-aware Machine Learning for Predicting Rare and Common Disease-Associated Non-Coding Variants. *Scientific Reports*.

Python Pakete

- <https://pypi.org/project/numpy/>
- <https://pypi.org/project/pandas/>
- <https://pypi.org/project/matplotlib/>
- <https://pypi.org/project/seaborn/>
- <https://pypi.org/project/scikit-learn/>
- <https://pypi.org/project/imblearn/>

Anhang

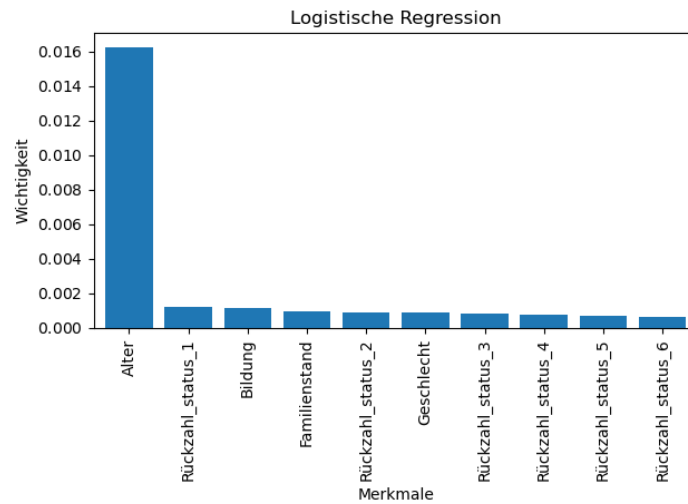


Abbildung 13: Merkmals-Wichtigkeit mit unbalancierten Daten

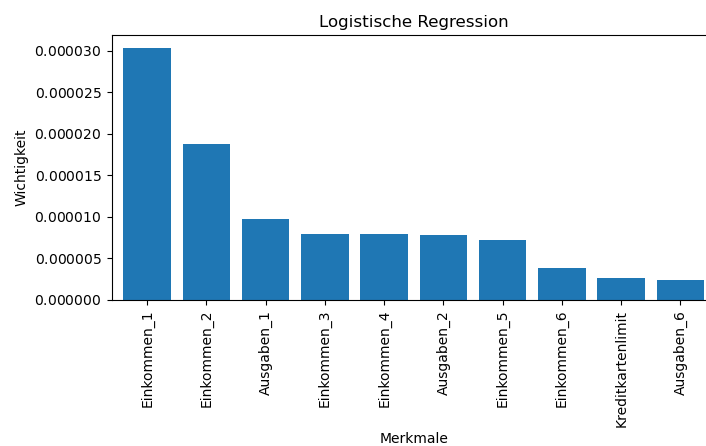


Abbildung 14: Merkmals-Wichtigkeit mit balancierten Daten